Solving Avatar Captchas Automatically

Mohammed Korayem[1,3], Abdallah A. Mohamed[2,4],
David Crandall[1], and Roman V. Yampolskiy[2]

[1] School of Informatics and Computing, Indiana University
Bloomington, Indiana USA
{mkorayem,djcran}@indiana.edu
[2] Computer Engineering & Computer Science, Speed School of Engineering
University of Louisville, Louisville, Kentucky USA
{aamoha04,roman.yampolskiy}@louisville.edu
[3] Department of Computer Science,
Fayoum University, Fayoum Egypt
[4] Department of Mathematics,
Menoufia University, Shebin El-Koom, Menoufia Egypt

Abstract. Captchas are challenge-response tests used in many online systems to prevent attacks by automated bots. Avatar Captchas are a recently-proposed variant in which users are asked to classify between human faces and computer-generated avatar faces, and have been shown to be secure if bots employ random guessing. We test a variety of modern object recognition and machine learning approaches on the problem of avatar versus human face classification. Our results show that using these techniques, a bot can successfully solve Avatar Captchas as often as humans can. These experiments suggest that this high performance is caused more by biases in the facial datasets used by Avatar Captchas and not by a fundamental flaw in the concept itself, but nevertheless our results highlight the difficulty in creating Captcha tasks that are immune to automatic solution.

1 Introduction

Online activities play an important role in our daily life, allowing us to carry out a wide variety of important day-to-day tasks including communication, commerce, banking, and voting [1, 9]. Unfortunately, these online services are often misused by undesirable automated programs, or "bots," that abuse services by posing as human beings to (for example) repeatedly vote in a poll, add spam to online message boards, or open thousands of email accounts for various nefarious purposes. One approach to prevent such misuse has been the introduction of online security systems called Captchas, or Completely Automated Public Turing tests to tell Computers and Humans Apart [1]. Captchas are simple challenge-response tests that are generated and graded by computers, and that are designed to be easily solvable by humans but that are beyond the capabilities of current computer programs [17]. If a correct solution for a test is received, it is assumed that a human user (and not a bot) is requesting an Internet service.[5]

[5] This paper is an expanded version of a preliminary paper [13] that appeared at the ICMLA Face Recognition Challenge [19].

Three main categories of Captchas have been introduced [4]. *Text-based Captchas* generate distorted images of text which are very hard to be recognized by state-of-the-art optical character recognition (OCR) programs but are easily recognizable by most humans. *Sound-based Captchas* require the user to solve a speech recognition task, while others require the user to read out a given sentence to authenticate that he/she is a human. Finally, *image-based Captchas* require the user to solve an image recognition task, such as entering a label to describe an image [9]. Other work has combined multiple of these categories into multi-modal Captchas [2], which can increase security while also giving users a choice of the type of Captcha they wish to solve.

The strength of a Captcha system can be measured by how many trials an attacking bot needs on average before solving it correctly [4]. However, there is a tension between developing a task that is as difficult as possible for a bot, but is still easily solvable by human beings. This is complicated by human users who may have sensory or cognitive handicaps that prevent them from solving certain Captchas. The best Captcha schemes are thus the ones which are easy for almost any human to solve but that are almost impossible for an automated program.

Recently, a novel image-based system was proposed called *Avatar Captcha* [6] in which users are asked to perform a face classification task. In particular, the system presents a set of face images, some of which are actual human faces while others are avatar faces generated by a computer, and the user is required to select the real faces. The designers of the scheme found that humans were able to solve the puzzle (by correctly finding all human faces) about 63% of the time, while a bot that randomly guesses the answers would pass only about 0.02% of the time.

In this paper, we consider how well a bot could perform against this Captcha if, instead of random guessing, it used computer vision algorithms to try to classify between human and avatar faces. Through experiments conducted on the human and avatar face images released by the authors of [6], we test a variety of modern learning-based recognition algorithms, finding that this task is surprisingly easy, with some algorithms actually *outperforming* humans on this dataset. While these results indicate that Avatar Captcha is not as secure as the authors had hoped, our results suggest that the problem may not be in the idea of Avatar Captcha, but instead in the way the avatar facial images were generated, allowing the recognition algorithms to learn subtle biases in the data.

2 Background and related work

As noted above, text-based Captchas are currently the most common systems on the web, and have been successfully deployed for almost a decade [1]. In order to increase the level of security against increasingly sophisticated OCR algorithms, text-based Captchas have had to increase the degree of distortion of the letters or numbers and hence may become so difficult that even humans are unable to recognize all of the text correctly. To address this problem, Captcha systems using image-based labeling tasks have been proposed [4, 7, 16]. No distortion is required for many of these tasks because humans can easily identify thousands of objects in images, while even state-of-the-art computer vision algorithms cannot perform this task reliably, especially when the set of possible classes is drawn from very large datasets [6]. While image-based

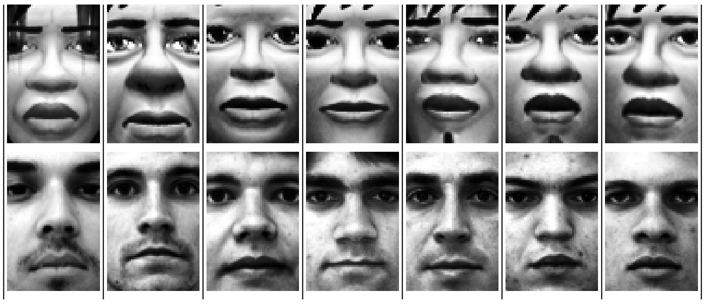


Fig. 1. Sample avatar faces (top) and human faces (bottom) from our dataset.

Captchas are still never completely secure, they are thought to widen the success rate gap between humans and non-humans.

Avatar Captcha. The authors of [6] proposed Avatar Captcha as a specific type of image-based task. In their approach, the system presents 12 images organized into a two-by-six matrix, with each image either a human face from a face dataset or a synthetic face from a dataset of avatar faces. The relative number of human and avatar faces and their arrangement is chosen randomly by the system. The user's task is to select all (and only) the avatar images among these 12 images by checking a checkbox under each avatar image. The user is authenticated as a human if he/she correctly completes the task, and otherwise is considered a bot. Using brute force attack, a bot has a success rate of 50% for each of the 12 images, since each image is either a human or avatar, so the probability of correctly classifying all 12 images is just $0.5^{12} = 1/4096$. Humans, on the other hand, were found to complete the task correctly about 63% of the time. In this paper, we show that a bot can achieve significantly higher performance than random guessing, and even outperform humans, using object recognition and machine learning.

3 Methods

We apply a variety of learning-based recognition approaches to the task of classifying between human and avatar faces. For data, we used a publicly-available dataset released by the authors of [6] as part of the Face Recognition Challenge held in conjunction with the International Conference on Machine Learning and Applications (ICMLA 2012) conference [19]. This dataset consists of 200 grayscale photos, split evenly between humans and avatars. The human dataset consists of frontal grayscale facial images of 50 males and 50 females with variations in lighting and facial expressions. The avatar dataset consists of 100 frontal grayscale facial images collected from the Entropia Universe and Second Life virtual worlds. All images were resampled to a uniform resolution of 50x75. Figure 1 shows sample images from the dataset.

Each of our recognition approaches follows the same basic recipe: we use a particular choice of visual feature which is used to produce a feature vector from an image, we learn a 2-class (human vs avatar) classifier using labeled training data, and then apply the classifier on a disjoint set of test images. We now describe the various visual features and classifiers that we employed.

3.1 Naïve Approaches

As baselines, we start with three simple approaches using raw pixel values as features.

Raw images. These feature vectors are simply the raw grayscale pixel values of the image, concatenated into a $50 \times 75 = 3750$ dimensional vector.

Summary statistics. As an even simpler baseline, we use a 1D feature that consists only of the mean grayscale value of the image. A second baseline represents each image as a vector of five dimensions, consisting of the maximum pixel value, the minimum pixel value, the average pixel value, the median pixel value, and the sum of all pixel values.

Grayscale histograms. This feature consists of a simple histogram of the grayscale values in the image. We tested different quantizations of the histogram, in particular testing histograms with 256, 128, 64, 32, 16, 8, 4, and 2 bins.

3.2 Histograms of Oriented Gradients (HOG)

Histograms of Oriented Gradients (HOG) features have become very popular in the recognition community for a variety of objects including people [5]. Computing these features consists of 5 stages: (1) global image normalization to reduce effect of changing illumination, (2) computing the image gradient at each pixel, (3) dividing the image into small 8x8 pixel cells, and then computing histograms over gradient orientation within each cell, (4) normalization of the histograms within overlapping blocks of cells, and (5) creating a feature vector, by concatenating all normalized histograms for all cells into a single vector. For the images in our dataset, this procedure yields a 2268 dimensional feature vector.

3.3 GIST

The GIST descriptor [15] was originally developed for scene recognition but has become popular for other recognition problems as well. This feature applies a series of filters to an image, each of which responds to image characteristics at different scales and orientations. The image is divided into a 4x4 grid of regions, and the average response of each filter is calculated within each region. This yields a descriptor that captures the "gist" of the scene: the orientation and scale properties of major image features at a coarse resolution, yielding a 960 dimensional vector.

3.4 Quantized feature descriptors

Another popular technique in recognition is to detect a sparse set of highly distinctive *feature points* in an image, calculate an invariant descriptor for each point, and then represent an image in terms of a histogram of vector-quantized descriptors. The Scale-Invariant Feature Transform (SIFT) [14] and Speeded-Up Robust Features (SURF) [3] are two commonly-used descriptors; we use the latter here. We use SURF to detect

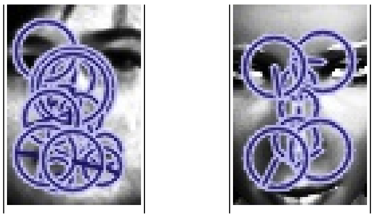


Fig. 2. Detected SURF features for a human face (left) and avatar face (right).

features points and calculate descriptors for each point, and then use k-means to produce a set of 50 clusters or "visual words." We then assign each descriptor to the nearest visual word, and represent each image as a histogram over these visual words, yielding a 50 dimensional feature vector. Figure 2 illustrates some detected SURF features.

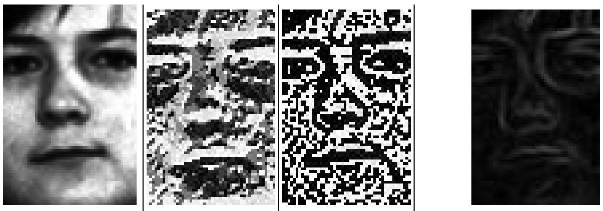
3.5 Local binary pattern-based features

Four-patch local binary pattern (FPLBP). The local binary pattern (LBP) descriptor examines each pixel in a small neighborhood of a central pixel, and assigns a binary bit depending on whether the grayscale value is greater than or less than that of the central pixel. The bits that represent the comparison are then concatenated to form an 8-bit decimal number, and a histogram of these values is computed. FPLBP is an extension to the original LBP where for each pixel in the image we consider two rings, an inner ring of radius r_1 and an outer one of radius r_2 (we use 4 and 5, respectively), each centered around a pixel [18]. T patches of size $s \times s$ (we use $s = 3$) are spread out evenly on each ring. Since we have T patches along each ring then we have $T/2$ center symmetric pairs. Two center symmetric patches in the inner ring are compared with two center symmetric patches in the outer ring, each time setting one bit in each pixels code based on which of the two pairs are more similar, and then calculate a histogram from the resulting decimal values.

Local Difference Pattern Descriptor. We also introduce a simple modification to the above approach which we call Local Difference Pattern. We divide the image into nxn $(3x3)$ windows and compute a new value for the center of each window based on the values of its neighbors. We compute the new value as the average of the differences between the center and all other pixels in the window (instead of computing the binary window and converting it into its decimal value as in LBP). We tried using both absolute and signed differences. Figure 3 illustrates this feature. Finally we compute a histogram for these new values.

3.6 Classifiers and Feature selection methods

For learning the models from each of the above feature times, we applied two different types of classifiers: Naïve Bayes [11,12], and LibLinear with L2-regularized logistic regression [8]. We used Correlation-based Feature Selection (CFS) [10] to reduce feature dimensionality.



original image | LBP | LDP | LDP-absolute difference

Fig. 3. Illustration of LBP and LDP features for a human face.

4 Results

Table 1 presents the results on the face-versus-avatar classification task for our simplest features (the Naïve features based on raw pixel values) and our simplest classifier (Naïve Bayes). All results presented here were evaluated using 10-fold cross-validation. The best classification rate obtained in this set of experiments is 93%, when raw grayscale pixel values concatenated into a vector are used as features. Interestingly, even much simpler techniques give results that are significantly better than random guessing (which would yield 50% accuracy). The 128-dimensional grayscale histograms achieve 92% accuracy, but even 4-dimensional histograms achieve almost 70% accuracy. Our simplest method, which encodes an image as a single dimension corresponding to its mean pixel value, gives an accuracy of 56%.

The fact that such simple recognition tools yield surprisingly high results suggests that there may be some unintended biases in the Avatar Captcha dataset that the classifiers may be learning. These biases could probably be removed relatively easily, by for example applying grayscale intensity and contrast normalization so that the histograms and summary statistics of human and avatar images would be identical. Figure 5 shows the most informative locations in the raw grayscale pixel features, and suggests that the key differences between avatars and humans are in the cheek lines and around the eyes.

We next tested more sophisticated techniques which may be much more difficult to guard against. Table 4 shows results for the more sophisticated features and classifiers that we tested. Each row of the table shows a different feature type, while the columns show results for classification using LibLinear, Naïve Bayes (NB), and Naïve Bayes with feature selection (NB+FS). Perfect recognition results (100% accuracy) are achieved by both the LibLinear classifier using raw pixel values, and the local difference pattern (LDP) descriptor using Naïve Bayes with feature selection. HOG features also produced excellent results (99% correct accuracy), while SURF and the local binary pattern variants all yielded accuracies above 95% for at least one of the classifiers. GIST and grayscale histogram features performed relatively poorly at around 90%, but this is still a vast improvement over the random baseline (50%). Figure 4 presents ROC curves for the different classifiers and features.

Table 1. Classification results using Naïve features and Naïve Bayes classifiers.

Method	Accuracy	Precision	Recall	F-measure
Pixel-values	**93%**	**93.2%**	**93%**	**93%**
Histograms(256-Bins)	89%	89.8%	89%	88.9%
Histograms(128-Bins)	92%	92.3%	92.%	92%
Histograms(64-Bins)	77%	77.3%	77%	76.9%
Histograms(32-Bins)	78%	78.2%	78%	78%
Histograms(16-Bins)	75%	75.1%	75%	75%
Histograms(8-Bins)	77%	77.9%	77%	76.8%
Histograms(4-Bins)	69%	69.1%	69%	69%
Histograms(2-Bins)	52%	52.1%	52%	51.7%
Average-mean-pixel	57%	57.4%	56%	53.8%
Avg_Min_Max_Sum_Median	61%	62.9%	61%	59.5%

Table 2. Classification accuracy using different features and classifiers, with feature dimensionality in parentheses.

Method	LibLinear	Naïve Bayes	Naïve Bayes + FS
Raw pixels	**100% (3750f)**	93% (3750f)	98% (54f)
Histogram	60% (256f)	**89% (256f)**	82% (24f)
GIST	84% (960f)	88% (960f)	**90% (24f)**
HOG	**99% (2268f)**	94% (2268f)	95% (44f)
FPLBP	94% (240f)	89% (240f)	**95% (26f)**
SURF codebook	**97% (50f)**	96% (50f)	94% (22f)
LDP (absolute differences)	94% (256f)	99% (256f)	**100% (61f)**
LDP (differences)	96% (256f)	98% (256f)	**99% (75f)**
LBP	**98% (256f)**	95% (256f)	98% (31f)

5 Discussion and conclusion

Our experimental results indicate that the current Avatar Captcha system is not very secure because relatively straightforward image recognition approaches are able to correctly classify between avatar and human facial images. For example, several of our classifiers achieve 99% accuracy on classifying a single image, which means that they would achieve $(0.99)^{12} = 88.6\%$ accuracy on the 12-face classification Captcha proposed in [6]. This results is actually better than the human performance on this task (63%) reported in [6]. Our classifiers work better than baseline even on surprisingly simple features, like summary statistics of an image. These results suggest that there may be substantial bias in the library of face images used in the current system, and that a new dataset without such biases would yield a much more secure system. Our work thus highlights the difficulty of creating image-based Captcha systems that do not suffer from easily-exploitable biases, and how to prevent such biases (and ideally to prove that they do not exist) is a worthwhile direction for future work.

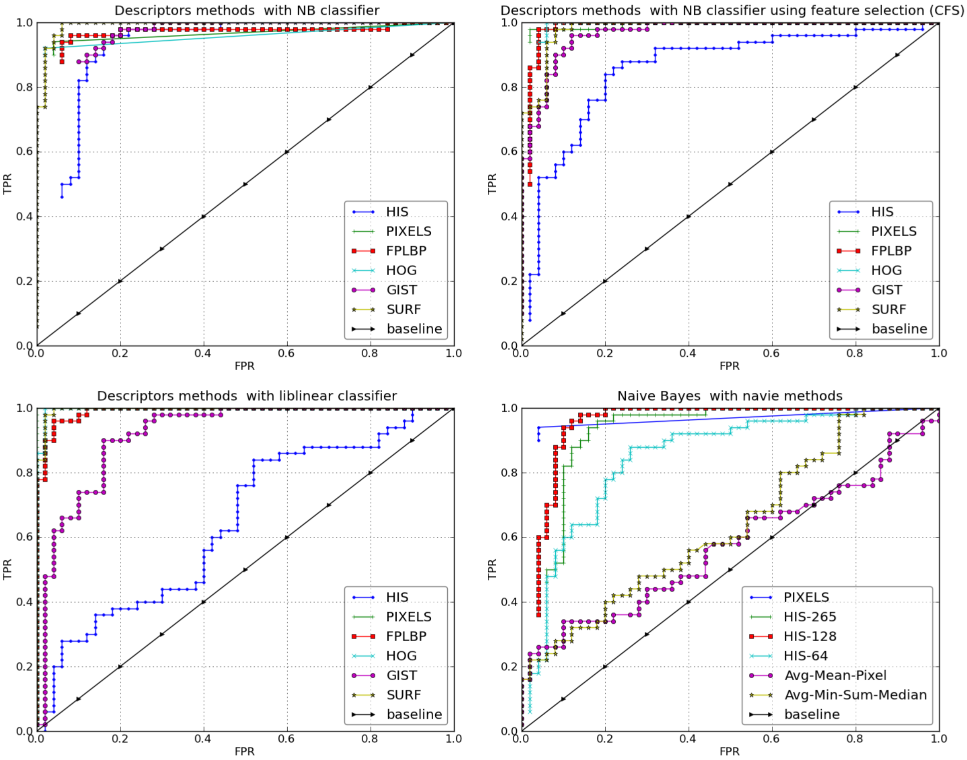
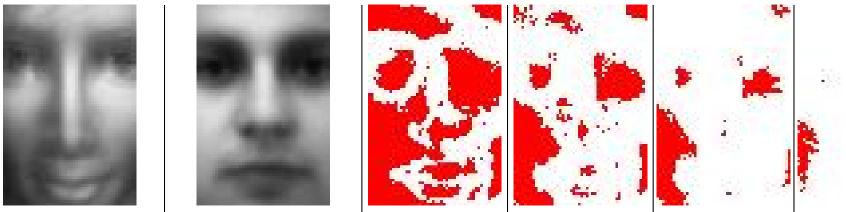


Fig. 4. ROC curves for the human versus avatar classification task. *Top left:* Naïve Bayes classifiers, *Top right:* feature selection and Naïve Bayes, *Bottom row:* LibLinear classifiers.



Mean avatar image | Mean human image | Top 2000 feat. | Top 1000 feat. | Top 500 feat. | Top 100 feat.

Fig. 5. *From left:* Mean face images, and positions of top features according to information gain.

References

1. L. Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using hard AI problems for security. In *Proceedings of the 22nd International Conference on Theory and Applications of Cryptographic Techniques*, pages 294–311. Springer-Verlag, 2003.

2. A. Almazyad, Y. Ahmad, and S. Kouchay. Multi-modal captcha: A user verification scheme. In *Proceedings of International Conference on Information Science and Applications (ICISA)*, pages 1–7. IEEE, 2011.
3. H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *European Conference on Computer Vision*, pages 404–417, 2006.
4. A. Chandavale, A. Sapkal, and R. Jalnekar. A framework to analyze the security of text based captcha. *International Journal of Computer Applications*, 1(27):127–132, 2010.
5. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, pages 886–893, 2005.
6. D. D'Souza, P. Polina, and R. Yampolskiy. Avatar captcha: Telling computers and humans apart via face classification. In *Proceedings of IEEE International Conference on Electro/Information Technology (EIT)*. IEEE, 2012.
7. J. Elson, J. Douceur, J. Howell, and J. Saul. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 366–374, 2007.
8. R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
9. H. Gao, D. Yao, H. Liu, X. Liu, and L. Wang. A novel image based CAPTCHA using jigsaw puzzle. In *Computational Science and Engineering (CSE)*, pages 351–356, 2010.
10. M. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.
11. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
12. G. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Uncertainty in Artificial Intelligence*, pages 338–345, 1995.
13. M. Korayem, A. Mohamed, D. Crandall, and R. Yampolskiy. Learning visual features for the avatar captcha recognition challenge. 2012.
14. D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, 1999.
15. A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.
16. L. Von Ahn, M. Blum, and J. Langford. Telling humans and computers apart automatically. *Communications of the ACM*, 47(2):56–60, 2004.
17. L. Wang, X. Chang, Z. Ren, H. Gao, X. Liu, and U. Aickelin. Against spyware using CAPTCHA in graphical password scheme. In *Proceedings of IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pages 760–767. IEEE, 2010.
18. L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *ECCV Workshop on Real-Life Images*, 2008.
19. R. Yampolskiy. ICMLA Face Recognition Challenge. http://www.icmla-conference.org/icmla12/.